

Docket No.: P-170

PATENT

#2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

JC914 U.S. PRO
09/750921

01/02/01

In re Application of

Kyoung Jin KANG, Ji Won YU and
Jae Hwon KWON

Serial No.: New U.S. Patent Application

Filed: January 2, 2001

For: SECURITY PROTOCOL STRUCTURE IN APPLICATION LAYER

TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Assistant Commissioner of Patents
Washington, D. C. 20231

Sir:

At the time the above application was filed, priority was claimed based on the following application:

Korean Patent Application No. 66105/1999, filed December 30, 1999.

A copy of each priority application listed above is enclosed.

Respectfully submitted,
FLESHNER & KIM, LLP



Daniel Y.J. Kim
Registration No. 36,186
Anthony H. Nourse
Registration No. 46,121

P. O. Box 221200
Chantilly, Virginia 20153-1200
703 502-9440
Date: January 2, 2001
DYK: AHN/cam

1C914 U.S. PRO
09/750921
01/02/01

2

대한민국특허청
KOREAN INDUSTRIAL
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Industrial
Property Office.

출원번호 : 특허출원 1999년 제 66105 호
Application Number

출원년월일 : 1999년 12월 30일
Date of Application

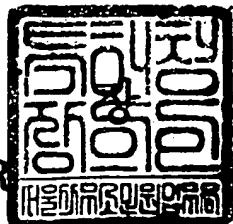
출원인 : 엘지정보통신주식회사
Applicant(s)



2000 09 월 18 일

특허청

COMMISSIONER



CERTIFIED COPY OF
PRIORITY DOCUMENT

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0020
【제출일자】	1999. 12. 30
【국제특허분류】	G06F
【발명의 명칭】	통신 프로토콜 운용 방법
【발명의 영문명칭】	Method for operating communication protocol
【출원인】	
【명칭】	엘지정보통신 주식회사
【출원인코드】	1-1998-000286-1
【대리인】	
【성명】	허용록
【대리인코드】	9-1998-000616-9
【포괄위임등록번호】	1999-047173-5
【발명자】	
【성명의 국문표기】	권재환
【성명의 영문표기】	KWON, Jae Hwan
【주민등록번호】	730201-1674025
【우편번호】	151-010
【주소】	서울특별시 관악구 신림동 10-392번지 301호
【국적】	KR
【발명자】	
【성명의 국문표기】	유지원
【성명의 영문표기】	RYU, Ji Weon
【주민등록번호】	680510-1696219
【우편번호】	407-040
【주소】	인천광역시 계양구 효성동 현대3차 102동 908호
【국적】	KR
【발명자】	
【성명의 국문표기】	강경진
【성명의 영문표기】	KANG, Kyoung Jin
【주민등록번호】	570105-1804322

【우편번호】 472-900
【주소】 경기도 남양주시 와부읍 덕소리 111-1 주공2차 207동 402호
【국적】 KR
【심사청구】 청구
【취지】 특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인 허용록 (인)
【수수료】
【기본출원료】 17 면 29,000 원
【가산출원료】 0 면 0 원
【우선권주장료】 0 건 0 원
【심사청구료】 1 항 141,000 원
【합계】 170,000 원
【첨부서류】 1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명에 따른 통신 프로토콜 운영 방법은, 서버가 클라이언트로부터 전송된 메시지를 받고 사용자 식별자에서 예비-마스터 시크리트를 검출하는 단계와, 서버가 클라이언트에게 서버 랜덤 값을 전송하기 위해서 ServerHello 메시지를 생성하는 단계와, 서버가 마스터 시크리트를 생성하는 단계와, 서버가 키 블록을 생성하는 단계와, 서버가 생성된 키 블록으로부터 최종적으로 암호, 해독 알고리즘과 MAC 알고리즘에서 사용할 키 값을 생성하는 단계와, 서버가 다음 메시지부터는 암호화해서 보낼 것이라는 사실을 알리기 위한 제 1 레코드를 생성하는 단계와, 서버가 클라이언트가 서버와 동일한 마스터 시크리트를 생성했다는 것을 확인할 수 있는 종료 메시지를 생성하는 단계와, 서버가 생성한 메시지를 집중화하여 클라이언트로 전송하는 단계와, 클라이언트가 서버와 동일한 방식으로 마스터 시크리트, 키 블록, 최종 암호와 키와 맥 키 값을 각각 계산하는 단계와, 서버가 보내는 메시지가 암호화되어 전송된 것을 확인하는 단계와, 클라이언트가 보내는 메시지가 서로 합의한 키 값으로 처리하여 전송할 것이라는 것을 알리는 제 2 레코드를 전송하는 단계를 포함한다.

【대표도】

도 4

【명세서】

【발명의 명칭】

통신 프로토콜 운용 방법{Method for operating communication protocol}

【도면의 간단한 설명】

도 1은 종래의 WAP 프로토콜 스택의 구조도.

도 2는 종래의 WTLS 프로토콜에서 핸드 쉐이크(Handshake) 프로토콜의 동작 절차를 나타낸 도면.

도 3은 본 발명에 따른 WAP 프로토콜 스택의 구조도.

도 4는 본 발명에 따른 SSLS 프로토콜의 핸드 쉐이크(Handshake) 절차를 보인 도면

<도면의 주요부분에 대한 부호의 설명>

101... 네트워크 계층 102... 전송 계층

103... 보안 계층 104... 트랜잭션 계층

105... 세션 계층 106... 비밀 세션 계층

107... 응용 계층

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<10> 본 발명은 통신 프로토콜에 관한 것으로서, 특히 현재 WAP 표준 상에서 응용 프로그램 계층에서 사용자를 인증하기에 적당한 통신 프로토콜 운용 방법에 관한 것이다.

<11> WAP(Wireless Application Protocol)은 휴대폰 등의 무선단말기로부터 인터넷 등의 콘텐츠를 효율적으로 이용하는 데 필요한 통신 프로토콜이다. 이와 같은 WAP은 이동통신 사업자, 정보 제공자, 단말기 제조업체가 이동통신망을 이용한 부가가치 통신사업에 대한 업계 표준을 제정하기 위해 에릭슨, 모토로라, 노키아와 언와이드 플래넷 등 4개의 회사가 중심이 돼 97년 6월에 결성되었다. 유럽의 디지털 이동통신 방식인 GSM을 기반으로 개발되었으나, 국내 표준인 CDMA는 물론 미래의 이동통신에도 적용될 수 있는 구조를 갖고 있다.

<12> 이와 같은 WAP 표준 상에서 전송되는 데이터에 대한 보안은 도 1에 도시된 바와 같이 전송(transport)계층(12)인 WDP(Wireless Datagram Protocol)(22)의 바로 상위 계층인 WTLS(Wireless Transport Layer Security)(23)에서만 이루어진다.

<13> WTLS 프로토콜은 SSL(Secured Sockets Layer)라고 불리던 산업계 표준인 TLS(Transport Layer Security) 프로토콜을 바탕으로 한 보안 프로토콜로서, 상대적으로 긴 지연 시간을 가진 저대역 폭을 가진 네트워크에 알맞게 최적화되어 있다. 이와 같은 WTLS(23)는 다음과 같은 기능을 제공하고 있다.

<14> 첫째, WTLS(23)는 단말기와 응용 프로그램 서버 사이에 전송되는 데이터가 바뀌거나 오염되지 않았다는 것을 확인하는 기능을 제공한다.

<15> 둘째, WTLS(23)는 단말기와 응용 프로그램 사이에서 전송되는 데이터가 중간에서 다른 불법 사용자에 의해 가로채임을 당하더라도 그 데이터의 내용이 해독되지 않도록 하는 기능을 제공한다.

<16> 셋째, WTLS(23)는 단말기와 응용프로그램의 서버 사이에서 사용자에 대한 인증 기

능을 제공한다.

<17> 도 2는 종래의 WTLS 프로토콜에서 핸드 쉐이크(Handshake) 프로토콜의 동작 절차를 나타낸 도면이다. 도 2를 참조하면, 먼저 클라이언트(client)와 서버 (Server)간 헬로우(Hello) 메시지를 교환해서 알고리즘에 합의하고 임의 값을 교환한다. 이어, 클라이언트와 서버가 예비-마스터 시크리트(pre-master secret)에 합의하기 위해 필요한 암호학의 인수들을 교환한다. 이어, 클라이언트와 서버가 서로를 인증하기 위해 필요한 인증(certificate)과 암호학의 정보를 교환한다. 그러면, 예비 마스터 시크리트와 교환할 임의의 값들로부터 마스터 시크리트를 생성한다. 이어, 레코드 계층에 보안 인수들을 제공한다(a1, b1 참조).

<18> 따라서, 클라이언트와 서버가 서로 동일한 보안 인수들을 계산해 내었다는 것과 핸드 쉐이크가 침입자의 개입없이 이루어졌다는 것을 확인한다(c1, d1 참조).

<19> 현재 단말기의 메모리 용량이나 중앙처리장치(CPU)의 프로세싱 파워는 WTLS가 달리 있는 인증서(certificate)를 사용한 사용자 인증이나 공개 키 생성 및 공개 키를 다루기에는 부적합하고, WTLS에서 제안한 프로토콜 포맷이 다소 복잡하기 때문에 데이터를 생성하고 해독하는 작업의 부하도 결코 무시할 수 없는 상태이다.

<20> 또한, 현재 WAP 표준 상에서는 WTLS는 기본적으로 전송 계층 바로 위에서 데이터 보안 기능을 제공하므로, 응용 프로그램 계층에서의 데이터 보안 기능을 제공하지는 못 한다. 또한, 현재 WAP 표준 상에서는 별도의 응용 프로그램 계층에서의 데이터 무결성(integrity), 데이터 보안, 사용자 인증에 대한 기능이 전혀 정의되어 있지 않았다. 따라서 응용 프로그램 계층에서 데이터 보안 기능을 제공하기 위해서는 별도의 수단이 정

의되어야 하며, 현재 단말기의 하드웨어적인 복잡성으로 인해서 인증서나 공개 키 연산과 관련된 PKI(Public Key Infrastructure)를 적용하기가 어려운 실정이다

【발명이 이루고자 하는 기술적 과제】

<21> 본 발명은 상기와 같은 문제점을 감안하여 창출된 것으로서, 현재 WAP 표준 상에서 응용 프로그램 계층에서 효율적으로 보안기능을 제공할 수 있는 통신 프로토콜 운용 방법을 제공함에 그 목적이 있다.

【발명의 구성 및 작용】

<22> 상기의 목적을 달성하기 위하여 본 발명에 따른 통신 프로토콜 운영 방법은,

<23> 클라이언트와 서버간 제공되는 통신 프로토콜 운용에서, 상기 서버가 상기 클라이언트로부터 전송된 메시지를 받고 사용자 식별자(id)에서 예비-마스터 시크리트를 검출하는 단계와;

<24> 상기 서버가 서버 랜덤(server random) 값을 생성하고, 그 값을 상기 클라이언트에게 전송하기 위한 서버 헬로우(Servo Hello) 메시지를 생성하는 단계와;

<25> 상기 서버가 검출한 예비-마스터 시크리트와 클라이언트 랜덤(client random), 서버 랜덤(server random)값을 바탕으로 마스터 시크리트를 생성하는 단계와;

<26> 상기 서버가 생성된 마스터 시크리트와 클라이언트 랜덤, 서버 랜덤 값을 바탕으로 키 블록(key block)을 생성하는 단계와;

<27> 상기 서버가 상기 생성된 키 블록으로부터 최종적으로 암호(encryption), 해독(decryption) 알고리즘과 MAC(Message Authentication Code) 알고리즘에서 사용할 키 값을 생성하는 단계와;

<28> 상기 서버가 다음 메시지부터는 암호화해서 보낼 것이라는 사실을 알리기 위한 제 1 ChangeCipherSpec 레코드를 생성하는 단계와;

<29> 상기 서버가 상기 클라이언트가 서버와 동일한 마스터 시크리트를 생성했다는 것을 확인할 수 있는 종료(Finished) 메시지를 생성하는 단계와;

<30> 상기 서버가 생성한 메시지를 집중화하여 상기 클라이언트로 전송하는 단계와;

<31> 상기 클라이언트가 자신이 가지고 있는 상기 예비-마스터 시크리트와 클라이언트 랜덤, 상기 서버로부터 받은 서버 랜덤 값으로부터 서버와 동일한 방식으로 마스터 시크리트, 키 블록, 최종 암호와 키와 맥 키 값을 각각 계산하는 단계와;

<32> 상기 클라이언트가 상기 레코드를 처리한 후, 상기 서버가 보내는 메시지가 암호화되어 전송된 것을 확인하는 단계와;

<33> 상기 클라이언트에서 종료 메시지를 검사해서 상기 서버와 동일한 마스터 시크리트를 생성한 것을 확인하고, 상기 클라이언트가 보내는 메시지가 서로 합의한 키 값으로 처리하여 전송할 것이라는 것을 알리는 제 2 ChangeCipherSpec 레코드를 전송하는 단계를 포함하는 점에 그 특징이 있다.

<34> 이와 같은 본 발명에 의하면, 현재 WAP 표준에서 제시하지 않은 응용 프로그램 계층에서의 특정 사용자에 대한 보안기능을 인증서나 공개키와 관련된 연산을 다루지 않고 간단한 공유 키를 기반으로 해쉬 연산으로 키를 생성 교환하는 방식으로 제공한다. 또한, WAP 상에서 뿐만 아니라 일반적으로 데이터 보안 기능을 지원하는 응용 프로그램 계층에서의 데이터 보안을 지원한다.

<35> 이하 첨부된 도면을 참조하면서 본 발명의 실시 예를 상세히 설명한다.

<36> 본 발명에서 제안하는 WAP 상의 응용 프로그램 계층의 보안 프로토콜은 인증서나 공개키와 관련된 연산을 다루지 않고, 사용자와 서버가 공유하는 비밀 키를 기반으로 이루어지며 도 3과 같이 세션(Session) 계층 위에서 동작하면서 응용 프로그램에게 비밀 세션(Secure Session) 인터페이스를 제공하기 때문에 본 발명에서는 SSLS(Secured Session Layer Security)라 명명한다.

<37> 도 3은 본 발명에 따른 WAP 프로토콜 스택의 구조도이다. 도 3을 참조하면, 도 1에 보인 종래의 구조와는 다르게 세션 계층(105) 바로 상위 계층에 비밀 세션 계층(106)이 형성되었음을 알 수 있다.

<38> 또한, 도 4는 본 발명에 따른 SSLS 프로토콜의 핸드 쉐이크 절차를 보인 도면이다. 도 4를 참조하면, SSLS 프로토콜에서 새로운 비밀 세션은 클라이언트와 서버가 각각 저장하고 있는 공유 비밀 값을 기반으로 이루어진다. 이 경우, 공유 비밀 값을 예비-마스터 시크릿으로 사용한다.

<39> 한편, SSLS 프로토콜의 핸드 쉐이크 절차를 설명하면 다음과 같다.

<40> 본 발명의 실시 예에서는 언급하는 프로토콜 데이터는 WTLS 표준에서 사용하는 프로토콜 기술 언어를 사용하였으며, PRF(Pseudo Random Function) 역시 WTLS 표준에서 사용하는 함수를 그대로 사용한다.

<41> 먼저, 클라이언트는 ClientHello 메시지를 서버에 전송한다. ClientHello는 클라이언트 랜덤 값 및 사용자 식별자를 각각 포함한다. ClientHello 메시지의 구조를 WTLS 표준에서 사용하는 프로토콜 기술 언어로 표현하면 다음과 같다.

<42> `struct{`

```

<43>     uint32 gmt_unix_time;

<44>     opaque random_bytes[12];

<45> } Random;

<46> opaque Identifier<1..2^8-1>;

<47> struct {

<48>     uint8 client_version;

<49>     Random random;

<50>     Identifier client_id;

<51> } ClientHello;

```

<52> 이어, 서버는 ClientHello 메시지를 받고서 사용자 식별자가 유효한지 검사한 후, 사용자 식별자에서 예비-마스터 시크리트를 검출한다. 이것은 서버 측에서 사용자 식별자에 대한 공유 예비-마스터 시크리트를 데이터 베이스로 관리하고 있기 때문에 가능하다. 이어, 별도의 서버 랜덤 값을 생성하여 ServerHello 메시지를 생성한다. 이 때의 ServerHello 메시지의 구조는 다음과 같다.

```

<53> Struct{
    uint8 server_version;
    Random random;
} ServerHello;

```

<57> 이어, 서버는 추출한 예비-마스터 시크리트와 클라이언트 랜덤, 서버 랜덤 값을 바탕으로 마스터 시크리트를 생성한다. 이어, 다시 생성된 마스터 시크리트와 클라이언트

랜덤, 서버 랜덤 값을 바탕으로 키 블록을 생성한다. 따라서, 이 키 블록으로부터 최종적으로 암호, 해독 알고리즘과 MAC 알고리즘에서 사용할 키(key) 값을 생성한다.

<58> 이때, 마스터 시크리트를 계산하는 방법은 다음과 같다.

```
<59> master_secret=PRF(pre_master_secret, 'master secret',
```

```
<60> ClientHello.random + ServerHello.random) [0..19];
```

<61> 또한, 키 블록을 계산하는 방법은 다음과 같다.

```
<62> key block=PRF(master_secret,
```

```
<63> expansion_label,
```

```
<64> Security Parameters.server_random +
```

```
<65> Security Parameters.client_random);
```

<66> 여기서, 키 블록으로부터 최종 키의 추출은 키 블록으로부터 순서대로 16 byte client MAC key, 16 byte client encryption key, 8byte client IV, 16byte server MAC key, 16byte server encryption key, 8byte server IV를 할당하는 방식을 이용한다.

<67> 이어, 서버는 다음 메시지부터는 암호화해서 보낼 것이라는 사실을 알리기 위한 ChangeCipherSpec 레코드를 생성한다.

<68> 이어, 서버는 실제 마스터 시크리트를 네트워크 상에서 전송하지 않고서도 클라이언트가 서버와 동일한 마스터 시크리트를 생성했다는 것을 확인할 수 있는 종료 메시지를 생성한다. 이 종료 메시지는 레코드 계층(record layer)에서 서버가 생성한 최종 암호와 키 값과 맥 키(MAC key)값으로 처리되어 전송되는 첫 번째 메시지이다. 이때의 종료 메시지의 구조는 다음과 같다.

<69> struct{

<70> opaque verify_data[12];

<71> } Finished;

<72> 여기서, 가변 데이터(verify_data)의 정의는 다음과 같다.

<73> PRF(master_secret, 'server finished', H(handshake_messages))[0..11];

<74> 여기서, 핸드 쉐이크 메시지(handshake_messages)는 ClientHello와 Server Hello 메시지를 집중화(concatenation)한 것이다.

<75> 이어, 서버가 생성한 ServerHello 메시지를 포함하는 핸드 쉐이크 레코드, ChangeCipherSpec 레코드, 종료 메시지를 포함하는 핸드 쉐이크 레코드는 네트워크 상에서 데이터를 교환하는 횟수를 줄이기 위해 집중화되어 한꺼번에 클라이언트에게 전송된다.

<76> 이어, 클라이언트는 ServerHello 메시지를 처리하고 나서 자신이 가지고 있는 예비-마스터 시크리트와 클라이언트 랜덤, 서버 랜덤 값으로부터 서버와 동일한 방식으로 마스터 시크리트, 키 블록, 최종 암호와 키와 맥 키 값을 각각 계산한다.

<77> 이어, 클라이언트는 ChangeCipherSpec 레코드를 처리한 후, 서버가 보내는 메시지가 암호화되어 전송될 것을 확인한다.

<78> 이어, 클라이언트는 종료 메시지를 검사해서 서버와 동일한 마스터 시크리트를 생성한 것을 확인한다. 이때 동일한 값을 생성하는 것을 확인한 후에 이후 클라이언트가 보내는 메시지가 서로 합의한 키 값으로 처리하여 전송할 것이라는 것을 알리는 ChangeCipherSpec 레코드를 전송한다.

<79> 이어, 핸드 쉐이크 과정이 성공적으로 완료되었으면, 이후 응용 프로그램 계층의 데이터를 암호화하여 서로 주고받는다.

【발명의 효과】

<80> 이상의 설명에서와 같이 본 발명에 따른 통신 프로토콜 운용 방법은, 현재 WAP 표준에서 제시하지 않은 응용 프로그램 계층에서의 특정 사용자에 대한 보안기능을 인증서나 공개 키 생성 및 공개 키를 이용한 키 교환을 다루지 않고 간단한 공유 키를 기반으로 해쉬 연산으로 키를 생성 교환하는 방식으로 제공한다. 따라서, 현재 단말기의 메모리 용량이나 중앙 처리장치의 능력으로도 적용할 수 있다. 본 발명에서 공유 키를 사용하는 것이 결국 사용자에 대한 암호이기 때문에 데이터 보안뿐만 아니라 부수적으로 간단한 형태로 특정 사용자에 대한 인증을 실행할 수도 있다. 또한, WAP 상에서 뿐만 아니라 일반적으로 데이터 보안 기능을 지원하는 응용 프로그램 계층에서의 데이터 보안을 지원한다.

【특허청구범위】**【청구항 1】**

클라이언트(client)와 서버(Server)간 제공되는 통신 프로토콜 운용에서,

상기 서버가 상기 클라이언트로부터 전송된 메시지를 받고 사용자 식별자 (id)에

서 예비-마스터 시크리트(pre-master secret)를 검출하는 단계와;

상기 서버가 상기 클라이언트에게 서버 랜덤(server random) 값을 전송하기 위해서

ServoHello 메시지를 생성하는 단계와;

상기 서버가 검출한 예비-마스터 시크리트와 클라이언트 랜덤(client random), 서

버 랜덤(server random) 값을 바탕으로 마스터 시크리트를 생성하는 단계와;

상기 서버가 생성된 마스터 시크리트와 클라이언트 랜덤, 서버 랜덤 값을 바탕으로

키 블록(key block)을 생성하는 단계와;

상기 서버가 상기 생성된 키 블록으로부터 최종적으로 암호(encryption), 해독

(decryption) 알고리즘과 MAC(Message Authentication Code) 알고리즘에서 사용할 키 값을

생성하는 단계와;

상기 서버가 다음 메시지부터는 암호화해서 보낼 것이라는 사실을 알리기 위한 제

1 ChangeCipherSpec 레코드를 생성하는 단계와;

상기 서버가 상기 클라이언트가 서버와 동일한 마스터 시크리트를 생성했다는 것

을 확인할 수 있는 종료(Finished) 메시지를 생성하는 단계와;

상기 서버가 생성한 메시지를 집중화하여 상기 클라이언트로 전송하는 단계와;

상기 클라이언트가 자신이 가지고 있는 상기 예비-마스터 시크리트와 클라이언트

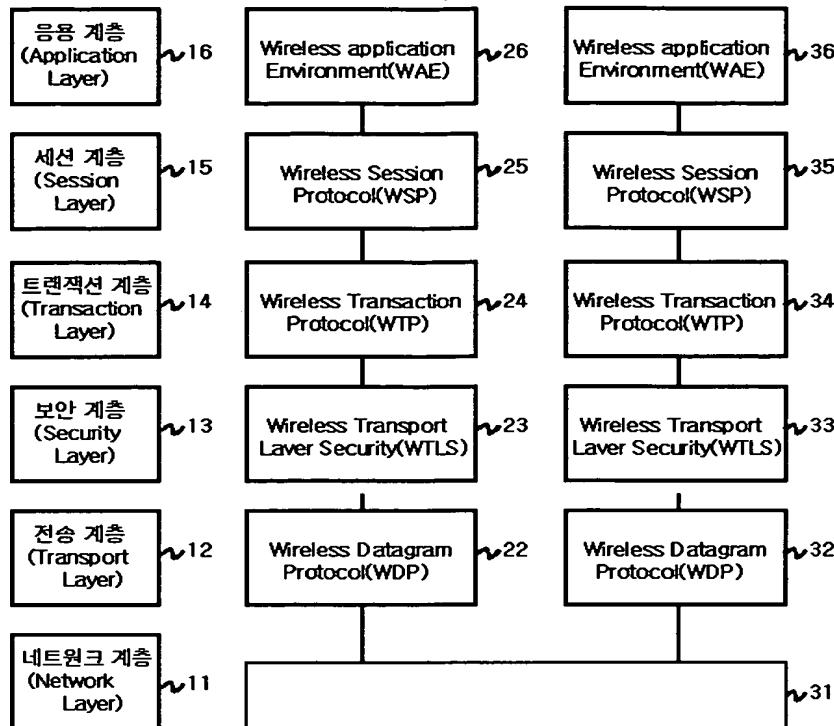
랜덤, 상기 서버로부터 받은 서버 랜덤 값으로부터 서버와 동일한 방식으로 마스터 시크리트, 키 블록, 최종 암호와 키와 맥 키 값을 각각 계산하는 단계와;

상기 클라이언트가 상기 레코드를 처리한 후, 상기 서버가 보내는 메시지가 암호화되어 전송된 것을 확인하는 단계와;

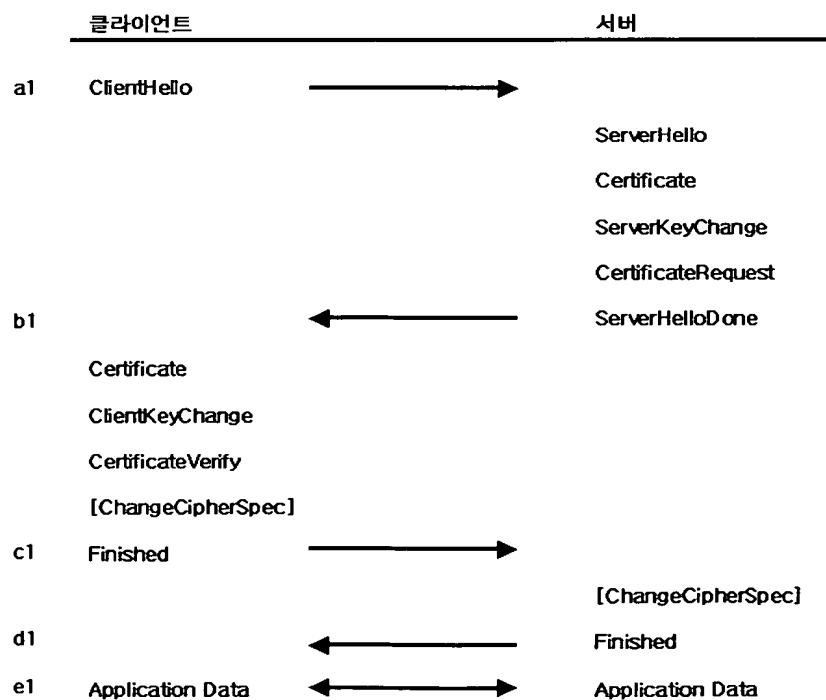
상기 클라이언트에서 종료 메시지를 검사해서 상기 서버와 동일한 마스터 시크리트를 생성한 것을 확인하고, 상기 클라이언트가 보내는 메시지가 서로 합의한 키 값으로 처리하여 전송할 것이라는 것을 알리는 제 2 ChangeCipherSpec 레코드를 전송하는 단계를 포함하는 것을 특징으로 하는 통신 프로토콜 운용 방법.

【도면】

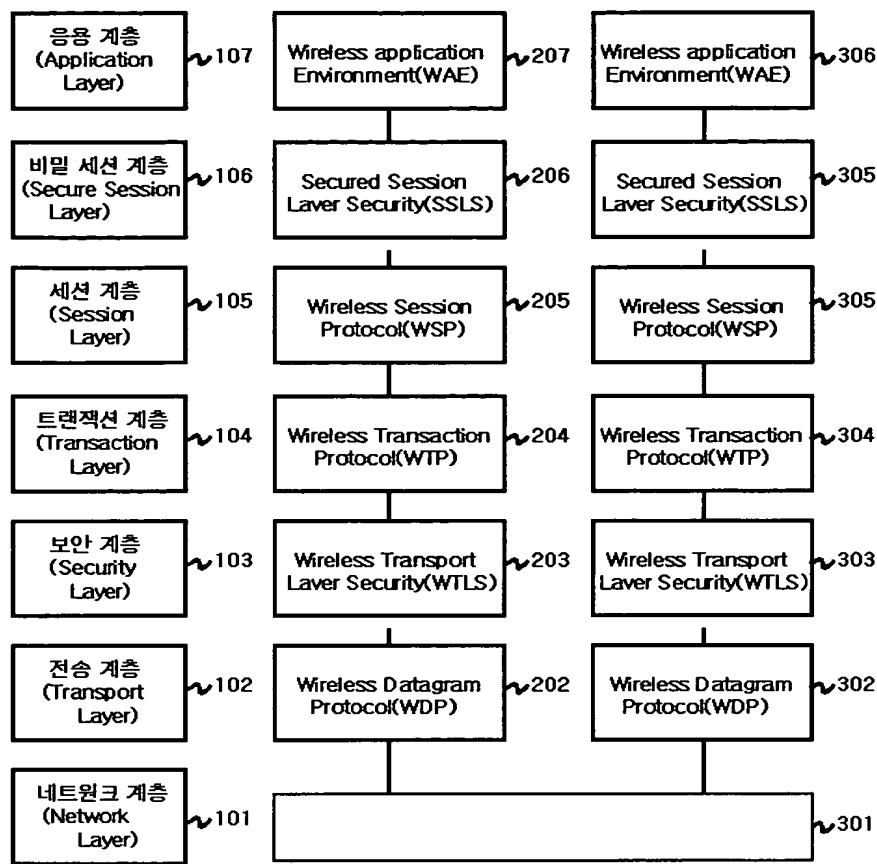
【도 1】



【도 2】



【도 3】



【도 4】

